Linking Microscopic and Macroscopic Models for Evolution: Markov Chain Network Training and Conservation Law Approximations

Roderick V. N. Melnik*

Mathematical Modelling and Computational Sciences,
Wilfrid Laurier University, 75 University Avenue West,
Waterloo, ON, Canada N2L 3C5 and MCI, SDU, Denmark, DK-6400

Abstract

In this paper, a general framework for the analysis of a connection between the training of artificial neural networks via the dynamics of Markov chains and the approximation of conservation law equations is proposed. This framework allows us to demonstrate an intrinsic link between microscopic and macroscopic models for evolution via the concept of perturbed generalized dynamic systems. The main result is exemplified with a number of illustrative examples where efficient numerical approximations follow directly from network-based computational models, viewed here as Markov chain approximations. Finally, stability and consistency conditions of such computational models are discussed.

Key words: Training dynamics, neural networks, complex dynamic systems, Markov chain approximation method, conservation law approximations.

1 Introduction

Many concepts and tools used in information theory, control, theory of approximation, dynamical systems and artificial intelligence are closely linked. Among such tools are neural networks. The success in using neural networks in many application areas is well documented in the literature. This includes, but not limited to, aeronautics, pattern and speech recognition (in particular in the context of various classification problems) [7, 11], optoelectronics, robotics and autonomous navigation, determining structure-property relationships in material science applications [40, 38], constructing dynamic observers [4], identification and control of complex biotechnological cycles, industrial processes, and nonlinear systems in general [41, 8], modelling complex dynamic systems and nonlinear phenomena such as hysteresis [17]. In many cases, a key to this success is kept by various associations of (computational) neural networks with *the atomistic approach* to molecular design aimed at determining some relationships between the properties of the structure (thermomechanical, electromagnetic etc) and the structure itself described at the microscopic (mesoscopic, and eventually macroscopic) level. Another set of tools and successful approaches to structure modelling is *the macroscopic approach* essentially based on the conservation law equations. In this contribution we developed a general framework for establishing a link between neural network concepts and numerical approximations of conservation laws.

*Tel.: +1-519-884-1970 (3662), Fax: +1-519-884-9738, E-mail: rmelnik@wlu.ca

One of the major objectives in neural network theory is the study of constructive approaches to the design of effective learning algorithms [15]. We observe that models for learning of neural networks and models for the evolution of dynamic systems are closely connected. The idea of using neural networks as components in dynamic systems is now well established in the literature (e.g., [30]). Models for the dynamic interactions between evolution and learning have been used by a number of authors who use evolutionary algorithms for the optimization of neural structures. It is often argued that artificial neural networks (ANN) and evolutionary algorithms (EA) can be efficiently combined together. Indeed, if ANN is interpreted as a model for biological nervous systems and the learning process, then EA can be interpreted as a model for biological evolution itself and the process of adaptation of large scales [37]. In this case, ANN becomes a useful mathematical tool for linking deterministic and probabilistic approaches in the approximation theory. This idea was developed in [24, 25] where the evolution was proposed to be modelled by a generalized equation with training/learning rules (algorithms) understood in a probabilistic sense. The problem of learning in neural network theory is formulated in terms of the minimization of an error function which is a function of adaptive parameters (weights and biases). In developing training/learning algorithms for ANN we have to deal with incomplete data, and an initial step of this procedure can be associated with the initializing of the weights in the network. Assigning these weights random values [12], leads us to a probabilistic framework [39]. Postprocessing of the available information is required for deriving deterministic models. For example, given a (training) set of input-output pairs $(u(i), y_i)$ $(i = 1, ..., n$ for n training/example subjects), we can "smooth" the data. The task is to construct a map which provides a good generalization (i.e. for given u that does not belong to this map should provide a reasonable estimate/prediction of the unobservable output). This is reducible to finding the best (functional) approximation to multivariate empirical (possible noisy, sparse, with some unobservable regions) data. However, if some additional "smoothing" constraints are used, we solve not the original, but a regularized problem. This provides a link between standard (Tichonov-like) approaches to deterministic, but ill-posed, problems and statistical (e.g. the Bayesian) approaches [9]. In this contribution we explore further this link between deterministic and probabilistic approaches in neural network theory by considering learning and evolution in an intrinsic dynamic connection.

The starting point of our discussion is a formalization of the learning/training process in terms of the minimization of an error function based on the neural network approach. In particular, starting with a minimal structure (no hidden layers), according to certain rules, new connections, neurons, and layers are added. Most commonly used are related to a (probabilistic) adaptation of the network weights, a procedure which is intrinsically coupled with the overall network size (number of neurons on each layer and the total number of layers). Since the size and topology of neural networks are closely connected with the required training time, in dealing with the training/learning problem we will have to construct *dynamic* models. Such models arise naturally in applications of ANN to optimal control problems in particular in those areas where system parameters are time varying [22]. The approach developed in this contribution has common features with the on-loan evolution approach (previously discussed in the literature in the context of game theory and control problems [2]) in a sense that our procedure can be interpreted as an on-line evolution algorithm applied to a certain time-dependent equation. The type of this equation is motivated by the fact that network architectures can be established by approximating the dynamic programming equations [5]. In this paper, we also use a dynamic equation that corresponds to a certain neural network architecture and is derived from the framework of generalized dynamic systems (GDS) developed in [24, 25].

In what follows, although we focus our attention on feed-forward-based networks complemented by backpropagation-like learning numerical procedures, the underlying procedure is the same for other architectures of neural systems where we have to define the connections between layers, the parameters (e.g., initial weights) and some learning rules. We demonstrate that these connections can be established by combining "forward evolution" (using neural networks) and " backward evolution" (using Markov chains associated with the original process). This combination ensures the minimization of uncertainty in the following sense. By processing information in response to discrete and continuous inputs, the ANN allows us to model dynamic systems. Since without a correction mechanism uncertainty may increase dramatically

over time, we use Markov chains to construct an appropriate mechanism where the architecture of the ANN depends on the cone of macroscopic events [25]. Although our approach is completely different in principle from the classical Gelenbe approach (see, e.g. [3]), we note that in both cases one has to exploit the idea of neural-network random structures. Since the original time-dependent models are discretized in our approach, one can interpreted the resulting scheme via interactions among neurons so it is possible to calculate probabilities of activation of network neurons in a way similar as it is done in the Gelenbe approach.

Finally, we note that when applying ANNs to complex dynamic systems, one of the most important issues is to stabilize the learning mechanism [16] (e.g., one has to avoid the unbounded growth of the adjustable parameter in ANN). We show in this paper that this non-trivial task can be linked directly to discretizing conservation law models, and that the ANN stability can be treated effectively through the stability of numerical approximations of conservation laws, rather than through the Lyapunov stability (e.g. [21]).

2 Systems Theoretic Framework for Constructing Neural Networks and Dynamic Training Rules

By applying neural networks to learning and identification of dynamic systems, we attempt to influence the behavior of these systems by some components built-in into the systems [30, 28]. Hence, the procedures for controlling such dynamic systems using ANN should be connected in one way or another with approximations of system dynamics. Moreover, since control, in addition to the requirements of fast and accurate, should ensure stability and robustness of the system, the stability of control is closely connected in such cases with the stability of numerical approximations of models describing the systems dynamics.

Recall that in most typical situations a nonlinear dynamic system can be described from a systems theoretic point of view by the following set of equations

$$\begin{cases} x(k+1) = f[x(k), u(k)], \quad f(0,0) = 0, \\ y(k) = h[x(k)], \quad h(0) = 0, \end{cases} \tag{2.1}$$

where $u(k)$, $y(k) \in \mathbb{R}^m$ and $x(k) \in \mathbb{R}^n$ are input, output, and state vectors, respectively, at discrete time k, and mappings $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are given functions, while control based on the state vector and control based only on input-output data needs not to be the same [28]. Model (2.1) provide adaptive means for implementing digital neuro-controlled systems and for further insight into the neural learning and adaptation. We note that instead of model (2.1), the dynamic system model can be determined through identification (e.g., with multi-layered neural networks) using the data obtained by a recurrence equation describing the output signal and taking into account the internal perturbations (which influence the output signal):

$$y(k+1) = \tilde{h}(x(k+1), y(k), u(k), e(k+1)). \tag{2.2}$$

New output $y(k+1)$, determined from the state vector $x(k+1)$ by using the given input data $[y(k), u(k)]$, will introduce error $e(k+1)$, representing internal perturbations at time $k+1$. For example, it can mimic the effects of synaptic time delay representing the "memory" of neurons which may differ from connection to connection. Mapping $\tilde{h}$ in (2.2) can be viewed as a perturbed function h. In principle, this equation, which combines the information given by model (2.1) is sufficient to describe the training procedure. From a numerical point of view such a nonlinear-control-framework consideration [28] can be viewed as an approximation to the first equation in (2.1), describing the evolution of states of the system, supplemented by an approximation of the "constitutive" law (the second equation in (2.1)), describing some properties of the system, control rules, and/or requirements imposed by the user. Hence, if the original description of the system is set by using control-theoretic approach, model (2.2) can be viewed as an approximation to the Hamilton-Jacobi-Bellman-type (HJB) equation describing some "energetic" characteristic of the system

such as cost or value function [24, 25, 26]. If the entire controlled process is modelled by the ANN, then the training/learning process for the neural network (using specific outputs with each of several inputs) can be associated with the dynamics of a certain conservation law model that describes the evolution of the nonlinear system. In this case, the inputs of ANN are associated with the initial conditions of the conservation law model. The description of the output-input training process will be understood here as a constructive backpropagation based on a process-associated Markov chain approximation. This approximation will be linked to numerical approximations of conservation laws.

Following [28], we consider a neural network as a conveniently parameterized class of nonlinear maps. In the next paragraphs we develop the methodology for linking this class of maps to conservation law approximations. In this context it is important to note that when a network is chosen to approximate a given mapping using IO data, the IO set is finite, whereas a conservation law defines an infinite set of input-output data. However, once such a conservation law model is approximated the process of neural network training and the approximate conservation laws can be described within the same general framework based on the concept of generalized dynamic systems [24, 25]. In what follows, our consideration will be pertinent to discrete-time (even though the original system might be continuous in time), which reflects the fact that most of complex systems are controlled by computers and therefore it is quite natural to consider approximations to complex dynamic systems as being discrete in time [31]. Moreover, any model, no matter how sophisticated it is, should account for uncertainties of the environment [24, 25] which are easier to deal with in the discrete time formulations [31]. Furthermore, the stability issues are much more transparent in the discrete space-time of events where dynamic systems evolve. Finally, in the context of neural networks the discrete-time consideration is the most natural way to proceed with the analysis because neural networks are proved to be universal approximators and any continuous function can be approximated arbitrarily well on a compact set with a (multilayer feedforward, radial basis function, or another) neural network (e.g., [14, 19, 6, 8]).

3 Approximating System Hamiltonians with Neural Networks

If the dynamic system under consideration is Hamiltonian, the problem of training neural networks with dynamic system rules governing the evolution of this system requires efficient procedures for approximating the Hamiltonian of such a system by a neural network. As a starting point, let T be a given set of times during which the network is trained, Σ is a state space of the dynamic system, U_T is a set of all permissible strategies for training, and X_T is the domain of definition of the system Hamiltonian assumed to be a compact Borel set [13]. Further constructions are based on the following fundamental property of applications of neural networks in the theory of approximation [14, 19, 6]:

Theorem 3.1 *If $H \in \mathbb{L}^1(X_T)$, then for any arbitrary small $\epsilon > 0$ there exists a network $\tilde{H}$ such that*

$$||H - \tilde{H}||_{\mathbb{L}^1(X_T)} < \epsilon. \tag{3.1}$$

For the description and control of complex dynamical systems via neural networks both feedforward (FNN) evolutionary networks (for the modelling of nonlinear input-output dynamics) and feedback evolutionary networks (for the implementation of training algorithms) are required. The link between those can be established on the basis of the Markov chain approximations as follows. First, recall that if H is a continuous function we can construct a uniform approximation (e.g., by using a FNN [14, 19]). Note also that if smoothness of the function H is measured in terms of its Fourier representation an actual estimation of the network performance can also be obtained (e.g., [6]). Therefore, it is natural to associate a FNN model with the approximation $\tilde{H}$ of the system Hamiltonian (for simplicity, with one layer of sigmoidal nodes or

units). In $\mathbb{R}^n$ this model can be implemented by the following functions

$$H_n(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i x_t(\mathbf{y} \cdot \mathbf{x} + \beta_i) + \alpha_0 \tag{3.2}$$

with $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n,\ \alpha_i, \beta_i \in \mathbb{R}$. The point that we are making here is that such models alone may be of limited applicability when applied in uncertain dynamic environments, because the quality of network performance is dependent strongly on its training capabilities with respect to the regularity of H and the structure of X_T. In considering computational models of system dynamics the structure of the network-approximator like (3.2) cannot be considered to be fixed, but rather it has to be adapted to the change of environment and internal perturbations. As a consequence, the most appropriate approximation in the framework of (3.1) can lead to a situation where activation function x_t may not be continuous, subject to regularity of H. Furthermore, even if this function is continuous, in order to account for the topological structure of X_T each network layer has to be characterized by both feedforward and feedback operators.

Remark 3.1 *In a special case, where $X_T \equiv \mathbb{R}^N$ and hence $H : \mathbb{R}^N \to \mathbb{R}^M$ ($N, M \in \mathbb{N}$), such operators may be interpreted through weight matrices [35].*

The situation we described above is of the same nature as in control theory where a regularity balance between control and value functions has to be achieved [23, 24]. In the context of our framework, this balance is achieved by simultaneous computational treatments of the "feedforward" and "backward" evolutionary processes, which can be effectively implemented by using along with feedforward approximators feedback networks such as block feedback networks (BFN). Below we explain this idea in detail.

Consider Hamiltonian H mapping X_T into $\mathbb{R}^M$ and its "computational equivalent", that is a Turing computable function H_T such that

$$H : X_T \to \mathbb{R}^M, \text{ and } H_T : \mathbb{N} \to \mathbb{N}. \tag{3.3}$$

Then, since BFN models have the same computing power as Turing machines (e.g., [35]), we conclude that there exists a BFN $\tilde{H}_T$ such that for any input n from a subset of $\mathbb{N}$ a finite number of network steps produces $H_T(n)$. Therefore, due to the Gödel numeration procedure (e.g., [10] and references therein), any Hamiltonian function $H : \mathbb{R}^N \to \mathbb{R}^M$ can be arbitrary well approximated by a network implementation of function H_T. Since both feedforward $\tilde{H}$ and feedback $\tilde{H}_T$ networks provide an approximation to H, the development of constructive algorithms for training requires further study of the connection between $\tilde{H}$ and $\tilde{H}_T$. This connection between the concepts pertinent to feedforward and feedback networks is often overlooked. Examples where this connection becomes transparent are provided by associative memory networks or recurrent neural networks, where feedforward connections should be supplemented by some "memory" neurons or synaptic time delay, which may vary from connection to connection. A natural way to establish such a connection is to construct a Markov chain associated with the system evolution in such a way that it reflects the process of network training on the problem specific information.

Now, we are in a position to formulate the problem of training in terms of approximating the system Hamiltonian in such a way that the necessity of applying both forward and backward dynamic rules is transparent. For this purpose we consider a relatively simple case, where we assume that

$$x_t : \mathbb{R} \to \Sigma \tag{3.4}$$

is a sigmoidal function, meaning that

$$\lim_{t \to -\infty} x_t = 0, \ \lim_{t \to +\infty} x_t = 1. \tag{3.5}$$

The function x_t here is the activation function for a neural network defined by its neurons as the following mapping

$$x_t \circ \mu : T \otimes \Sigma \otimes U_T \to \Sigma, \tag{3.6}$$

where $\mu : T \otimes \Sigma \otimes U_T \to \mathbb{R}$ is known as the decision maker (DM) function [25] such that it has to be adjusted (trained) so that the neural network (3.6) leads to an approximation of H. If the network (3.6) is trained with some dynamic rules by using a dynamic system whose Hamiltonian is H, then the process of approximation of function H in terms of (3.1) can be seen as the construction of a training strategy for a new network $\tilde{H}_n^\epsilon$ depending on the arbitrarily small positive parameter ϵ and arbitrarily large number of sigmoidal nodes of the associated network n (e.g., (3.2)) so that

$$||H - \tilde{H}_n^\epsilon||_{\mathbb{L}^1(X_T)} \to \min, \tag{3.7}$$

where $H \in \mathbb{L}^1(X_T)$. The main difficulty in the solution of problem (3.7) stems from a priori unknown character of dependency of the network on parameters which determine the function μ. In the most general setting, the problem of constructing the mapping μ is intrinsically connected with the definition of dynamic rules in singular stochastic control problems [24], and in interpreting (3.7) one expects that

$$\text{if } \epsilon \to 0^+ \text{ and } n \to \infty \text{ then } \tilde{H}_n^\epsilon \to H. \tag{3.8}$$

However, for the constructive solution of (3.7) by using neural networks we need additional information. For example, if FNNs are applied to the solution of the problem, we need information on a coupling rule between ϵ, n, and the topology specified by X_T. This coupling rule will determine conditions for the system stability. If BFNs are applied to the approximation of H on an arbitrary set X_T, we need additional information on the network dimension and architecture [35]. This information can be made available only in *a sequential manner*, and the appropriate tool for the analysis of the network performance in such situations is a family of Discrete Markovian Decision Processes (DMDP). In this case a model for the training process, written in terms of the decision maker function μ, depends on Markov Chain parameters with possible discontinuities that are dependent on values of the sigmoidal function x_t. If the functional dependency of μ is chosen a priori, this may lead to the underestimation of possible irregularities of the function H which can happen if, e.g., smoothness of H is measured in terms of its Fourier representation [6, 27]. In the next section we aim at constructing a model that gives an approximation to μ with respect to some learning rules determined from the Markovian property of the process (x_t, μ).

4 Modelling Dynamics of Network Training

From a constructive approximation point of view, the training process of neural networks describing dynamic systems evolution can be seen as the solution to the following problem. We have to construct a model (possibly, a hierarchy of models [23]) for the decision maker function μ in such a way that a neural network approximates the Hamiltonian of a dynamic system evolution, while the dynamics of this evolution is described by the process of network training.

Any two sets of input-output data in the training process can be represented by the mathematical model of a dynamic system that couples two space-time events of the system evolution, e_n and e_{n+1}, by a function of the perturbed velocity v_ϵ and the system Hamiltonian or its approximation H:

$$e_{n+1} = H(v_\epsilon, e_n), \quad n = 0, 1, ... \tag{4.1}$$

The perturbed velocity is introduced to account for the changing environment and/or internal perturbations [25]. Then, if we specify a sequence of events $(e_0, e_1, ...)$ by temporal evolution and formalize the dynamics of the system by a discrete-time model, we obtain the following two equations [25]

$$\begin{cases} x_{t+1} = H_1(v_1, x_t), \\ h_{\tau+1} = H_0(v_0, h_\tau), \end{cases} \tag{4.2}$$

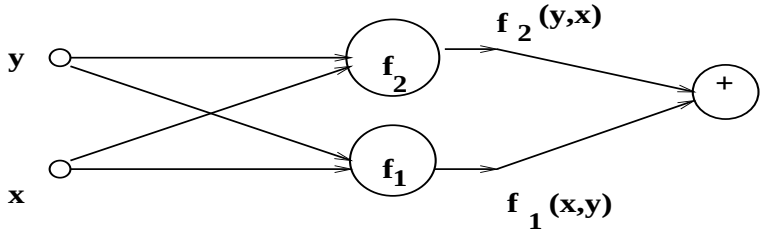

Figure 1: A typical unit of network architecture

where H_1 is an approximation to H and $v_1 = \lim_{\varepsilon \to 0^+} v_\varepsilon$, H_0 is an operator for sequential corrections of such an approximation needed due to system external/internal perturbations and v_0 is the function characterizing the rate of such perturbations. The dynamics of the system described by (4.2) operates on two coupled temporal scales, t and τ with spatial trajectories described by x_t and h_τ, respectively. Since perturbations (internal and/or external) is an intrinsic part of any dynamic system, we associate the class of neural networks with built-in training capability (determined by models for μ) as Infinite Length Perturbed Markov Chains (ILPMC) [25]. Since the form of the functional dependency of μ cannot be a priori fixed, the weights and the network structure may change which leads to a situation much more complicated than traditionally dealt with (see Fig. 1). Some algorithms dealing with such a situation are known (e.g., cascade-correlation, pruning algorithms, etc). However, our approach is different from the previous developed since it is based on the Markovian character of the process (x_t, μ), associated with both system dynamics and the process of training, and the concept of generalized dynamic systems (with appropriate assumptions on dynamic stochastic rules [25]). In our approach, by using conditional probabilities of the Markov chain, the velocity function between two macroscopic events in the evolution of the generalized dynamic system is introduced as a measure of changes which take place on the microscopic level with respect to the macroscopic behavior of the system. This velocity function provides a link between microscopic and macroscopic models for the evolution of dynamic systems.

Similar to model (3.7), models (4.1) and (4.2) are interpreted in terms of the limit $\epsilon \to 0^+$ and $n \to \infty$, and the connection between them can be analyzed when learning rules for the system evolution in response to perturbations are introduced. In particular, in the limit (3.8) both sequences $(x_0, x_1, ...)$ and $(h_0, h_1, ...)$ merge. As an important part of the system dynamics, perturbations can be formalized from the very beginning of the modelling process as a decision making process with limited information available.

Remark 4.1 *Since perturbed and unperturbed models might give rise to qualitatively distinct types of descriptions of system behavior for any arbitrary $\epsilon > 0$, in [25] it was already emphasized that the perturbation parameter alone cannot be an appropriate characteristic of the model's uncertainty, and this fact results in two informational sequences in model (4.2) considered on* two different time scales.

The learning rules are introduced into the model by the equation written in terms of function μ (Section 3). In particular, following [25, 26], we arrive at

Proposition 4.1 *If $H \in \mathbb{L}^1(X_T)$, then the process of training/learning can be associated with the following equation:*

$$(1 + v_1) \left[\frac{\partial \mu}{\partial x} + \frac{1}{v_1} \left(\frac{\partial \mu}{\partial t} + f_0 \right) \right] = 0, \tag{4.3}$$

where, in the context of neural networks, v_1 is the velocity of information transmission between neurons, and f_0 is a training/learning goal defined by a priori knowledge/assumptions on X_T and the function H.

This process can be approximated by an appropriately constructed Markov chain associated with the evolution of the dynamic system under consideration. Since function μ is required to be adapted to a priori given information, the practical implementation of training procedures may lead to approximations of the network-activator functions by a piecewise-deterministic stochastic process. Such non-diffusion stochastic models have been previously studied in theory of DMDP and in theoretical physics (see [25] and references therein). Such models are multiscale models for the evolution of dynamic systems described by (4.2) and, as pointed out in [25], can be interpreted by a system of coupled differential equations on two scales:

$$\begin{cases} \dot{h}(\tau) = v_0(\tau, h, \mu), \\ \dfrac{\partial \mu}{\partial t} + v_1(t, x, \mu)\dfrac{\partial \mu}{\partial x} = 0. \end{cases} \tag{4.4}$$

We conclude this section with the following observation.

Remark 4.2 *Both parts of the perturbed velocity functions v_0 and v_1 inherit their dependency on the decision-maker function μ. If two events between which GDS evolution has to be studied are specified (e.g., two sets of input-output data are entered), then a pair of functions $(h(\tau), \mu(t, x))$ gives the solution to the training problem under consideration.*

Next, we describe a procedure that allows us to approximate this pair.

5 Network-Based Computational Models for Dynamics as Markov Chain Approximations.

Although the perturbed system dynamics x_t^ϵ might not be governed by the Markovian property (and the function x_t^ϵ might not be sigmoidal), the pair of functions $(h(\tau), \mu(t, x))$ does possess the Markovian property [25]. Therefore, the key idea here is based on the construction of a Markov chain approximation simultaneously with an approximation of the system dynamics (which depends on Markov chain parameters). This allows us to guarantee system stability and to derive stability conditions in explicit form. The Markov chain will play in our constructions the role of a "training/learning" rule for the system dynamics considered in the case where the perturbed system's velocity is replaced by its approximation (function v_1) in the macroscopic (or decision maker's) frame of reference.

More precisely, we will approximate the pair of functions $(h(\tau), \mu(t, x))$ (which describes the process of GDS evolution and possesses the Markovian property) by a pair of discrete functions

$$(h(\tau), \mu(t, x)) \to (\xi_n^{\tau h}, \mu_n^{\tau h}), \tag{5.1}$$

where $\xi_n^{\tau h}$ is an associated (with the microscopic frame of reference) Markov Chain state.

First, we consider model (4.1) describing the evolution of dynamic systems in discrete space-time of events. Let an elementary space-time cell be c_{ij} and the complete spatio-temporal region, where the systems dynamics is studied, be $\bar{G}$ such that two events

$$e_j,\ e_{j+1} \in c_{ij} \equiv [x_i, x_{i+1}] \otimes [t^j, t^{j+1}] \subset \bar{G} \tag{5.2}$$

of system evolution are governed by the process (x_t, μ_t). Then we specify these events by two pairs of discrete functions

$$e_j = (\xi_j^{\tau h}, \mu(x_i, t^j)),\ \ e_{j+1} = (\xi_{j+1}^{\tau h}, \mu(x_{i+1}, t^{j+1})), \tag{5.3}$$

where $\xi_j^{\tau h} = x_i^j$ and $\xi_{j+1}^{\tau h} = x_{i+1}^{j+1}$ are states of the associated Markov Chain.

Next, we consider the perturbed generalized dynamic system whose training/learning dynamics can be described by the following equation, obtained from (4.3) or (4.4) under appropriate assumptions

$$\frac{\partial \mu}{\partial t} + v_1(t,x,\mu)\frac{\partial \mu}{\partial x} = \tilde{f}_0(t,x,\mu) \tag{5.4}$$

with the approximation of the initial condition in the DM-time scale

$$\mu(x,t)|_{t=t_0} = \delta(\epsilon), \tag{5.5}$$

where $\delta(\epsilon)$ is a set of initial conditions with ϵ dependent on the approximation of the function v_0 in the coupled system (4.2) (or (4.4)).

To preserve basic macroscopic features of the system, the values of jumps $\Delta\xi_j^{\tau h} = \xi_{j+1}^{\tau h} - \xi_j^{\tau h}$ of this chain should be subordinated to the corresponding approximation of system-environment boundaries. This can be done via establishing the connection between the GDS and perturbed GDS.

Definition 5.1 *Let (5.2), (5.3) be two subsequent macroscopic events of GDS evolution that are taking place with probability 1. Then the GDS velocity function between the macroscopic events e_j and e_{j+1} can be defined in an elementary space-time cell $c_{ij} \subset \bar{G}$ as*

$$v(t,x) = \lim_{\tau \to 0} \frac{E^{\tau h|(x_i,\mu^j)}\Delta\xi_j^{\tau h}}{\tau}, \tag{5.6}$$

where the numerator under the limit in (5.6) is the velocity of the Markov Chain (v_{MC}) between two subsequent macroscopic events.

We observe that

Remark 5.1 *If* $\lim\limits_{\epsilon \to 0^+} v_\epsilon = v_1$ *then*

$$\lim_{\epsilon \to 0^+,\, n \to \infty} v(t,x) = v_1, \tag{5.7}$$

and together with (4.1) this defines an Infinite Length Unperturbed Markov Chain (ILUMC).

Since perturbations cannot be ignored, (5.7) can be fulfilled only approximately, which leads to the consideration of an ILPMC, the concept already mentioned in Section 4.

Before formulating consistency conditions for the Markov chain associated with the dynamics of training rule (5.4), (5.5) we recall [25] that

Definition 5.2 *The state space in the initial moment of observation in the macroscopic frame of reference is defined as*

$$\Xi(i;0) = \{x_i, i = 0,1,2,...,N; \quad N = 2n, \quad n = \lceil (T - t_0)/h \rceil\}, \tag{5.8}$$

and subsequently the cone of macroscopic events of system evolution is defined by a set of macroscopic events as a mapping from the minimal resolution set for the identification of all macroscopic events relevant to the system evolution in the limit $n \to \infty$ to $\to \Xi(i;j)$, where

$$\Xi(i;j) = \{(x_i, t_j), \quad i = \overline{k, 2n-k},\ j = k,\ k = \overline{0,n}\}. \tag{5.9}$$

Now we are in a position to formulate

Proposition 5.1 The consistency conditions *(local and global, respectively) of the Markov Chain* $\xi_n^{\tau h}$, $n < \infty$ *with the Markov process* $(h(\tau), \mu(t,x))$*, defined by the mathematical model of GDS evolution (5.4), (5.5), are:*

$$E_n^{\tau h|(x_i,\mu^j)}\Delta\xi_j^{\tau h} = v_1(x_i, t^j, \mu^j)\tau + o(h+\tau) \tag{5.10}$$

$$cov_n^{\tau h|(x_i,\mu^j)}\Delta\xi_j^{\tau h} = o(h+\tau). \tag{5.11}$$

Using these ideas and following [25], a simple approximation to (5.4), (5.5) can be constructed and in the next sections we generalize this approximation to a large class of schemes for approximating the dynamics of training rules in neural network applications. Let us first highlight the main steps for the construction of the scheme proposed in [25].

- In the cone of macroscopic events we introduced a floating grid:

$$\omega_{\tau h}^{\triangle} = \{(x_i, t_j^{\tau_{j-1}}),\ i = \overline{k, 2n-k},\ j = k,\ k = \overline{0, n}\}, \tag{5.12}$$

  where $t_j^{\tau_{j-1}} = t^{j-1} + \tau_{j-1}$ when $j > 1$, $t_j^{\tau_{j-1}} = t^0 + \tau$ when $j = 1$, and $t_j^{\tau_{j-1}} = t^0$ when $j = 0$, where we constructed the following discrete scheme (based on upwind approximations with flux limiters [33])

$$\begin{aligned} d_i^{j+1} &= d_i^j\{1 - \frac{\tau}{h}[|v| + v^-\gamma_4 - v^+\gamma_1]\} + \frac{\tau}{h}d_{i-1}^j\{[v^+(1+\gamma_2) + v^-\gamma_4]\} + \\ &\frac{\tau}{h}d_{i+1}^j\{[v^-(1-\gamma_3) - v^+\gamma_1]\} + \frac{\tau}{h}d_{i-2}^j\{[-v^+\gamma_2]\} + \frac{\tau}{h}d_{i+2}^j\{v^-\gamma_3\}, \end{aligned} \tag{5.13}$$

  where d is a discrete function approximating function μ on grid (5.12).

- It is easy to see that the sum of all coefficients near the unknown function on the right-hand side (5.13) gives the unity. This fact allows us associate these coefficients with transition probabilities of a Markov Chain, provided those coefficients are nonnegative:

$$\left[\begin{array}{l} 1 - \frac{\tau}{h}(|v| + v^-\gamma_4 - v^+\gamma_1) \geq 0, \quad \gamma_2 \leq 0, \quad \gamma_3 \geq 0, \\ v^+(1+\gamma_2) + v^-\gamma_4 \geq 0, \quad v^-(\gamma_3 - 1) + v^+\gamma_1 \leq 0. \end{array}\right. \tag{5.14}$$

- The consistency conditions (local and global, respectively) of the Markov Chain (defined by time-transitions of the discrete scheme (5.13)) with the process $(h(\tau), \mu(t,x))$ (defined by the model (5.4),(5.5)) couple flux limiters of the scheme

$$\tau[v^-(1 - \gamma_4 + \gamma_3) - v^+(1 + \gamma_1 - \gamma_2) - v] = o(\tau + h), \tag{5.15}$$

$$\tau\{h[v^+(1 - \gamma_1 - 3\gamma_2) + v^-(1 + \gamma_4 + 3\gamma_3)] - \tau v_{MC}^2\} = o(\tau + h), \tag{5.16}$$

  where the Markov chain velocity in this case is determined as $v_{MC} = v^-(1-\gamma_4+\gamma_3) - v^+(1+\gamma_1-\gamma_2)$, and o is the Landau symbol. Using the idea of probabilistic characteristics the term $o(\tau + h)$ in (5.15) and (5.16) can be eliminated. The nonnegativeness of covariance leads to an additional stability condition

$$\frac{\tau}{h} \leq \frac{v^+(1 - \gamma_1 - 3\gamma_2) + v^-(1 + \gamma_4 + 3\gamma_3)}{[v^-(1 + \gamma_3 - \gamma_4) - v^+(1 + \gamma_1 - \gamma_2)]^2}, \tag{5.17}$$

  which can be satisfied under appropriate choice of the flux limiters. As a result, we have arrived to the stable Markov chain approximation of the training process.

- More precisely, if the interpolation interval τ is such that conditions (5.14) and (5.17) are satisfied and the transition probabilities of the Markov Chain $(\xi_n^{\tau h},\ n<\infty)$ are taken in the form

$$p^{\tau h}[x_k^j, x_i^{j+1}|d(x_k^j,t^j)] = \begin{cases} 1-\frac{\tau}{h}[|v|+v^-\gamma_4 - v^+\gamma_1], & k=i, \\ \frac{\tau}{h}[v^+(1+\gamma_2)+v^-\gamma_4], & k=i-1, \\ \frac{\tau}{h}[v^-(1-\gamma_3)-v^+\gamma_1], & k=i+1, \\ -\frac{\tau}{h}(v^+\gamma_2), & k=i-2, \\ \frac{\tau}{h}(v^-\gamma_3), & k=i+2, \\ 0, & \text{otherwise}, \end{cases} \tag{5.18}$$

$\forall j=\overline{0,n-1}$ and $i=\overline{j,N-j}$ ($\gamma_2=0$ for $i=j$ and $\gamma_3=0$ for $i=N-j$), then the Markov Chain approximation of the process $(h(\tau),\mu(x,t))$ is stable, and

$$d(x_i^{j+1},t^{j+1}) = \sum_k p^{\tau h}[x_k^j, x_i^{j+1}|d(x_k^j,t^j)]d(x_k^j,t^j). \tag{5.19}$$

A numerical procedure like (5.19) is an explicit (evolution forward) stabilization procedure where the DM-function is a stabilizing factor subject to the velocity of the system. Moreover, when $n\to\infty$ the velocity of the Markov Chain converges to the velocity of the process in the sense of the Markov theorem (e.g., [25]). This puts the approach used here on a rigorous theoretical basis. In the next section we show that this approach can be generalized to an important class of conservation-law-based models for the training process of neural networks.

6 Network-Based Probabilistic Approach to Conservation Law Approximations

The network-based methodology developed in the previous sections can be applied to conservation law approximations. In what follows we demonstrate this on a number of practical examples where we show that many efficient finite difference schemes for the approximation of conservation law equations follow directly from our technique.

Consider a model analogous to that of training/learning dynamics for perturbed GDS, but with $\mu\to u$, $v_1\to a$, and $\tilde{f}_0\equiv 0$:

$$\frac{\partial u}{\partial t} + a(t,x,u)\frac{\partial u}{\partial x} = 0, \tag{6.1}$$

where u is some physical quantity of interest, and a is the velocity related to the transport of that quantity. With the Cauchy initial condition

$$u(x,0) = u_0(x), \quad x\in\mathbb{R} \tag{6.2}$$

this equation gives a model of a physical system without dissipation in the form easily amenable to a nonlinear (quasi-linear) conservation law. Indeed if $a(u)=F'(u)$, then model (6.1), (6.2) is equivalent to

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0, \quad u(x,0)=u_0(x), \quad x\in\mathbb{R}. \tag{6.3}$$

This model and model (5.4), (5.5) describing the dynamics of network training belong to the same class of mathematical models, and such models provide building blocks for many mathematics applications in science and engineering. One of the sources of difficulties for numerical solutions of such problems is that in

many practical situations the solution to (6.3) might not be continuous, and one has to deal with possible discontinuities across certain curve $x = \sigma(t)$. In other words, if

$$\lim_{x\to\sigma(t)^-} u(x,t) = u_{\text{left}}, \quad \lim_{x\to\sigma(t)^+} u(x,t) = u_{\text{right}} \quad \text{and} \quad u_{\text{left}} \neq u_{\text{right}}, \tag{6.4}$$

one should be able to select a physical meaningful solution from a set of all possible solutions (since two or more classical characteristics in this case might pass the same point). In particular, considering the Cauchy problem (6.1), (6.2), we note that its solution should be understood in the generalized sense, where both the Rankine-Hugoniot condition [18]

$$[u]\frac{d\sigma}{dt} = [f], \tag{6.5}$$

where $[\cdot]$ denotes a jump across $x = \sigma(t)$ (i.e. $[u] = u_{\text{right}} - u_{\text{left}}$), and the entropy condition

$$a(u_{\text{right}}) < \frac{d\sigma}{dt} < a(u_{\text{left}}), \tag{6.6}$$

meaning that the entropy increases as material crosses discontinuity, should be satisfied. Under these circumstances the development of constructive procedures for efficient numerical schemes for the solution of (6.1), (6.2) becomes a very important task in theory and practice of conservation laws, as well as in a number of related fields. The stability of such schemes is at the heart of the success in achieving this task. Although viscosity solutions could provide some insight into these problems, they would not provide details on the solution stability. Hyperbolic partial differential equation (PDE) models are models for information propagations while probabilistic approaches imply diffusion. However, since such models need to be solved numerically, we will demonstrate that the stability conditions, derived in our approach in a straightforward manner, coincide with the stability conditions typical for numerical approximations of hyperbolic PDEs.

6.1 Neural networks in approximating conservation law models

There is an intrinsic analogy in the information flow pattern for neural networks used in applications (e.g., Fig. 1) and information flows represented by stencils of numerical approximations of conservation laws (e.g., Fig. 2). Consider first some basic classical schemes for numerical approximations of (6.1). We start from the forward centered Euler scheme

$$u_i^{j+1} = u_i^j - \frac{\lambda}{2}(u_{i+1}^j - u_{i-1}^j). \tag{6.7}$$

The flow of information for these schemes is given in Fig. 2 (a) and it can be seen that in the terminology of neural networks the stencil for this approximation (as well as for an improved approximation $u_i^j \to \frac{1}{2}(u_{i+1}^j + u_{i-1}^j)$) does not contain any hidden layers. The situation becomes more involved for the Leap-Frog scheme (see Fig. 2 (b)), where several layers of network architecture are coupled

$$u_i^{j+1} = u_i^{j-1} - \frac{\tau}{h}(F_{i+1}^j - F_{i-1}^j). \tag{6.8}$$

This coupling is sequential in nature, and hence our consideration is pertinent to such networks architectures where the layers are *sequentially* linked. From a numerical point of view such architectures can be represented by the predictor-corrector-type schemes. One of the most effective practical tool in numerical approximation of conservation laws that reflects this architecture is the Lax-Wendroff scheme

$$u_i^{j+1} = u_i^j - \lambda(F_{i+1/2}^{j+1/2} - F_{i-1/2}^{j+1/2}), \quad \text{where} \quad F_{i+1/2}^{j+1/2} = \frac{1}{2}au_{i+1}^j + \frac{\lambda}{2}a^2(u_{i+1}^j - u_i^j). \tag{6.9}$$

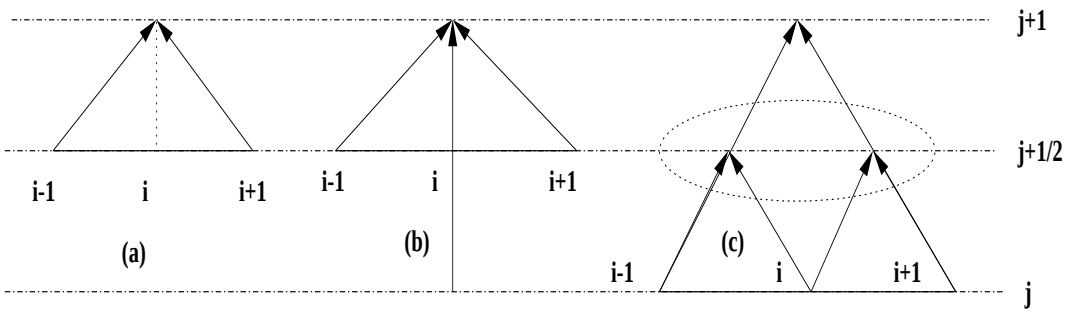

Figure 2: Network architectures and information flow chart for numerical approximations of conservation laws: (a) forward centered Euler, (b) leap-frog, (c) Lax-Wendroff.

As it is obvious from Fig. 2 (c), this scheme includes "hidden" layer $j + 1/2$ where one has to determine fluxes before moving to the next network layer. In what follows we will show that this scheme and many other effective schemes for numerical approximations of conservation laws can be derived from the general probabilistic approach based on the association of the conservation law models with Markov chain training processes for some neural network architectures in a way similar to that described in Section 5. Moreover, the interpretation/association of the stencils of these schemes with neural network architectures will allow us to deal with the stability conditions in an efficient manner by using the consistency conditions of Markov chains with the original process.

In the spirit of Section 5, the architecture of neural networks is associated with the cone of macroscopic events. From the numerical point of view this is equivalent to the definition of the following points

$$[(x_i, t_0^{\tau_{-1}}), i = 0, ..., 2n] \to [(x_i, t_1^{\tau_0}), i = 1, ..., 2n-1] \to [(x_i, t_2^{\tau_1}), i = 2, ..., 2n-2] \to ... \to$$

$$[(x_i, t_{n-1}^{\tau_{n-2}}), i = n-1, n, n+1] \to [(x_i, t_n^{\tau_{n-1}}), i = n] \tag{6.10}$$

in the cone of macroscopic events, where the approximation of the evolution of dynamic systems described by the conservation law takes place. An important point to emphasize is that in this general framework the stability conditions for the associated schemes can be obtained in the explicit form. Consider, for example, scheme (5.18), (5.19) for the solution of homogeneous problem (5.4), (5.5). First note that if the following condition

$$v^+(1 - \gamma_1 - 3\gamma_2) + v^-(1 + \gamma_4 + 3\gamma_3) = [v^-(1 + \gamma_3 - \gamma_4) - v^+(1 + \gamma_1 - \gamma_2)]^2 \tag{6.11}$$

is met, then the Courant-Friederichs-Lewy-type (CFL-type) stability condition (5.17) is satisfied [25]. Since either v^- or v^+ is zero while the other is non-zero, we consider 2 cases. Our aim in this example is to derive explicit expressions for the flux limiters. In its turn, as follows from our discussion in Section 5, this will define stability conditions of the associated scheme and will complete the definition of the transition probabilities of the associated Markov chain.

1. If $v^- = 0$ and $v^+ \neq 0$ then from (6.11) we have

$$1 - \gamma_1 - 3\gamma_2 = (v^+)^2(1 + \gamma_1 - \gamma_2)^2. \tag{6.12}$$

This leads to a quadratic equation with respect to γ_1

$$v^+\gamma_1^2 + [2v^+(1 - \gamma_2) + 1]\gamma_1 + [v^+(1 - \gamma_2)^2 + 3\gamma_2 - 1] = 0. \tag{6.13}$$

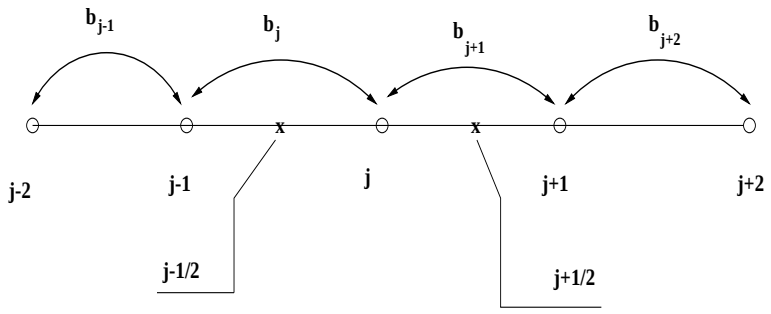

Figure 3: Flux limiters in the probabilistic approach for conservation laws

In a particular case where $\gamma_2 = 0$ the solution to (6.13) can be easily found

$$\gamma_1^{1,2} = \frac{-(2v^+ + 1) \pm \sqrt{8v^+ + 1}}{2v^+}. \tag{6.14}$$

Since γ_1 should be non-positive (see (5.14)) we take the sign "minus" in (6.14), leading to

$$\gamma_1 = -1 - \frac{\sqrt{8v^+ + 1} + 1}{2v^+}. \tag{6.15}$$

A similar reasoning leads to the expression for γ_1 in the general case of (6.13)

$$\gamma_1^{1,2} = \frac{-[2v^+(1 - \gamma_2) + 1] \pm \sqrt{8v^+ + 1 - 16v^+\gamma_2}}{2v^+}, \tag{6.16}$$

and finally to

$$\gamma_1 = -1 + \gamma_2 - \frac{\sqrt{8v^+ + 1 - 16v^+\gamma_2} + 1}{2v^+}. \tag{6.17}$$

2. Along the same vein we can obtain the expression for γ_4 in the case where $v^+ = 0$ and $v^- \neq 0$ as a solution to the equation

$$1 + \gamma_4 + 3\gamma_3 = v^-(1 + \gamma_3 - \gamma_4)^2 \tag{6.18}$$

which can be written in a form easily amenable to its solution with respect to γ_4

$$v^-\gamma_4^2 - [2v^-(1 + \gamma_3) + 1]\gamma_4 + [v^-(1 + \gamma_3)^2 - 1 - 3\gamma_3] = 0. \tag{6.19}$$

The solution to the last equation is determined as

$$\gamma_4 = 1 + \gamma_3 + \frac{\sqrt{8v^- + 1 + 16v^-\gamma_3} + 1}{2v^-} \tag{6.20}$$

(the sign "plus" was taken due to the non-negativeness of γ_4, see (5.14)). In the case $\gamma_3 = 0$ (6.20) is reduced to the expression obtained in [25].

This probabilistic approach based on the association of conservation law approximations and the neural network training process is amenable to the treatment of dissipative systems by using the framework of perturbed generalized dynamic systems developed in [25] (see also references therein). However, to make our basic concepts transparent to the reader, we concentrate in this paper on dynamic systems without

dissipation. Since the representation (4.1) of dynamic system (and its consequences) from a numerical analysis point of view is an explicit scheme, we recall [33, 20] that a family of explicit schemes for the solution of (6.1), (6.2) can be written as

$$u_j^{n+1} = u_j^n - \lambda(h_{j+1/2}^n - h_{j-1/2}^n), \tag{6.21}$$

where $h_{j+1/2}^n = h(u_j^n, u_{j+1}^n)$ is the numerical flux that can model hidden layers in the network architecture associated with the given scheme.

We propose the following general approximation of the flux at $x_j + 0.5\tau$ and $x_j - 0.5\tau$, respectively

$$h_{j+1/2} = \sum_{k=j-1}^{j+2} b_k(\tau, h, v_k)u_k = b_{j-1}u_{j-1} + b_j u_j + b_{j+1}u_{j+1} + \tilde{b}_{j+2}u_{j+2}, \tag{6.22}$$

$$h_{j-1/2} = \sum_{k=j-1}^{j} b_k(\tau, h, v_k)u_{k-1} = \tilde{b}_{j-1}u_{j-2} + b_j u_{j-1} + b_{j+1}u_j + b_{j+2}u_{j+1}, \tag{6.23}$$

where all coefficients in (6.22) and (6.23) are velocity-dependent approximations of the flux limiters, for example, $b_{j-1} \equiv b_{j-1}(\tau, h, v_{j-1})$, $\tilde{b}_{j-1} \equiv \tilde{b}_{j-1}(\tau, h, v_{j-2})$, etc. Then, the general family of the explicit schemes can be written as follows

$$\begin{aligned} u_j^{n+1} &= u_j^n - \lambda[-\tilde{b}_{j-1}u_{j-2} + (b_{j-1} - b_j)u_{j-1} + (b_j - b_{j+1})u_j + (b_{j+1} - b_{j+2})u_{j+1} + \tilde{b}_{j+2}u_{j+2}] \\ &= u_j^n - \lambda\left[-\tilde{b}_{j-1}u_{j-2} - \sum_{k=j-1}^{j+1}(b_k - b_{k+1})u_k + \tilde{b}_{j+2}u_{j+2}\right]. \end{aligned} \tag{6.24}$$

Note that this technique is applicable to the case where $b_m \equiv b_m(\tau, h, \psi(v_{m-1}, v_m))$, e.g. $\psi(v_{m-1}, v_m) = (v_{m-1} + v_m)/2$.

6.2 Special Cases and Examples

Now, we demonstrate our methodology on a number of examples and show that most efficient schemes for numerical approximations of conservation laws become just special cases of our general network-based approach.

Example 1. First of all, it is easy to confirm that by choosing the flux limiters in the forms

$$b_{j-1} = 0, \quad b_{j-1} - b_j = -\frac{1}{2}a, \quad b_j - b_{j+1} = 0, \quad b_{j+1} - b_{j+2} = \frac{1}{2}a, \quad b_{j+2} = 0, \tag{6.25}$$

we obtain the forward centered Euler scheme for which the numerical flux is defined as

$$h_{j+1/2} = \frac{1}{2}a(u_{j+1} + u_j). \tag{6.26}$$

Example 2. If we take now

$$\left[\begin{array}{l} b_{j-1} = 0, \quad -\lambda(b_{j-1} - b_j) = \frac{1}{2} + \frac{\lambda}{2}a, \quad 1 - \lambda(b_j - b_{j+1}) = 0, \\ -\lambda(b_{j+1} - b_{j+2}) = \frac{1}{2} - \frac{\lambda}{2}a, \quad b_{j+2} = 0, \end{array}\right. \tag{6.27}$$

we arrive at the following result

$$b_{j+1} = \frac{1}{2}a - \frac{1}{2\lambda}, \quad b_j = \frac{1}{2}a - \frac{1}{2\lambda}. \tag{6.28}$$

Therefore, we confirm that under the above choice of the flux limiter coefficients, we obtain the Lax-Friedrichs scheme

$$u_j^{n+1} = \frac{1}{2}(u_{j+1} + u_{j-1}) - \frac{\lambda a}{2}(u_{j+1} - u_{j-1}), \tag{6.29}$$

which can be written in the form (6.21) with

$$h_{j+1/2} = \frac{1}{2}\left[a(u_{j+1} + u_j) - \frac{1}{\lambda}(u_{j+1} - u_j)\right]. \tag{6.30}$$

Example 3. By choosing the flux limiters in the general architecture (6.21) – (6.23) as

$$b_{j-1} = b_{j+2} = 0, \quad b_{j-1} - b_j = -\frac{a}{2} - \frac{\lambda}{2}|a|, \quad b_j - b_{j+1} = |a| \tag{6.31}$$

we also confirm that

$$b_j = \frac{1}{2}a + \frac{|a|}{2}, \quad b_{j+1} = \frac{1}{2}a - \frac{|a|}{2}. \tag{6.32}$$

It can be seen that this leads to the (upwind) Euler uncentered scheme, representable in the general form (6.21) with

$$h_{j+1/2} = \frac{1}{2}\left[a(u_{j+1} + u_j) - |a|(u - u_j)\right]. \tag{6.33}$$

Example 4. Taking

$$b_{j-1} = 0, \quad b_j = \frac{1}{2}a + \frac{\lambda a^2}{2}, \quad b_{j+1} = \frac{1}{2}a - \frac{\lambda a^2}{2}, \quad b_{j+2} = 0 \tag{6.34}$$

in the general formulation (6.21) – (6.23) leads to the one of the most popular schemes for the numerical approximations of conservation law (6.3), the Lax-Wendroff scheme, discussed briefly earlier in this section, where

$$h_{j+1/2} = \frac{1}{2}[a(u_{j+1} + u_j) - \lambda a^2(u_{j+1} - u_j]. \tag{6.35}$$

Example 5. Finally, we note that the family of schemes proposed in [25] and briefly reviewed in Section 5 is also a subset of the general representation given by (6.21) – (6.23). Indeed, if we take

$$\left[\begin{array}{l} b_{j-1} = v, \quad b_{j-1} - b_j = -v^+(1+\gamma_2) - v^-\gamma_4, \quad b_j - b_{j+1} = |v| + v^-\gamma_4 - v^+\gamma_1, \\ b_{j+1} - b_{j+2} = -v^-(1-\gamma_3) + v^+\gamma_1, \quad b_{j+2} = v^-\gamma_3, \end{array}\right. \tag{6.36}$$

(as before, γ_j are flux limiters of the scheme determined from the stability and consistency conditions), we can determine flux coefficient b_{j+1} in two different ways. First, "moving from the left" (see Fig. 3) it is easy to obtain the following chain of relationships:

$$b_j = v^+(1+\gamma_2) + v^-\gamma_4, \quad b_{j+1} = b_j - |v| - v^-\gamma_4 + v^+\gamma_1 = v^+(1 + 2\gamma_2 + \gamma_1) - |v|. \tag{6.37}$$

On the other hand, "moving from the right" (see Fig. 3) we get that

$$b_{j+1} = v^-\gamma_3 - v^-(1-\gamma_3) + v^+\gamma_1 = v^-(2\gamma_3 - 1) + v^+\gamma_1. \tag{6.38}$$

Therefore, by equating two expressions for b_{j+1} we arrive at the conclusion that

$$v^+(1+2\gamma_2) + v^-(1-2\gamma_3) = |v|, \tag{6.39}$$

which can be satisfied by setting $\gamma_2 = \gamma_3 = 0$. Having these flux limiters, γ_1 and γ_4 can be controlled by satisfying the stability conditions [25].

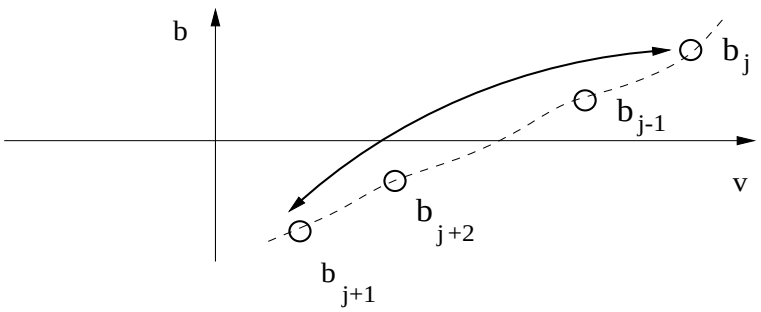

Figure 4: A schematic representation of flux limiters as functions of velocity

7 Stability and Consistency Requirements for Network-Based Approximations of Conservation Laws

In the previous section we derived numerical approximations for conservation laws using the general network-based methodology. As we have discussed in Section 4 this methodology originates from the probabilistic foundations of the dynamics. Indeed, having approximate initial data (e.g., as a result of measurements) we attempt to extrapolate this data further in time. This procedure is a subject of a probabilistic error. Therefore, it is important to discuss further stability and consistency requirements for the schemes obtained earlier in this paper, in particular, for the general family of approximations (6.21) – (6.23).

The probabilistic approach to conservation laws allows us to explain a number of important phenomena related to numerical approximations of these models in a straightforward way. First, it is easy to check that the sum of all coefficients near the unknown function in every scheme discussed in the previous section is 1. However, this fact alone might not be sufficient to interpret those coefficients as transition probabilities. Indeed, the nonnegativity requirement plays a key role in the possibility of such an interpretation. From a numerical point of view this requirement leads to the stability conditions of the scheme. If we take the forward Euler/centered scheme this requirement leads to $\lambda a/2 \le 0$, which can be satisfied only if $a < 0$ ($\lambda = \tau/h$). However, since we use here a forward scheme with a stencil depicted in Fig. 2, this should not come as a surprise, because for $a > 0$ the explicit scheme on this stencil is known to be unstable.

It is easy to check that coefficients in all schemes considered in Sections 5 and 6 will be nonnegative subject to the Courant-Friedrichs-Lewy-type stability conditions $|\lambda a| < 1$ (as one would expect for the models based on hyperbolic PDEs [33]) and therefore, all these schemes can be interpreted in a probabilistic sense. Naturally, however, that satisfying those stability conditions is subject to the appropriate choice of the flux limiters. For example, for the scheme (5.18), (5.19) discussed in Section 5, such fluxes (γ_1 and γ_4) should be chosen as to satisfy the following conditions

$$\frac{\tau}{h}(|v| + v^-\gamma_4 - v^+\gamma_1) \le 1, \quad \frac{1}{2}v^+ + v^-\gamma_4 \ge 0, \quad \frac{1}{2}v^- + v^+\gamma_1 \le 0. \tag{7.1}$$

Moreover, one has to satisfy the consistency conditions, which in the case of this scheme has the form

$$\tau\left[v^-(3/2 - \gamma_4) + v^+(3/2 + \gamma_1) - v\right] = o(\tau + h), \tag{7.2}$$

where, as before, o denotes the Landau symbol.

In our general case of the family (6.21) – (6.23) the nonnegativity of coefficients require

$$\left[\begin{array}{l} \lambda b_{j-1} \ge 0, \quad \lambda(b_j - b_{j-1}) \ge 0, \quad 1 - \lambda(b_{j+1} - b_j) \ge 0, \\ \lambda(b_{j+2} - b_{j+1}) \ge 0, \quad -\lambda b_{j+2} \ge 0. \end{array}\right. \tag{7.3}$$

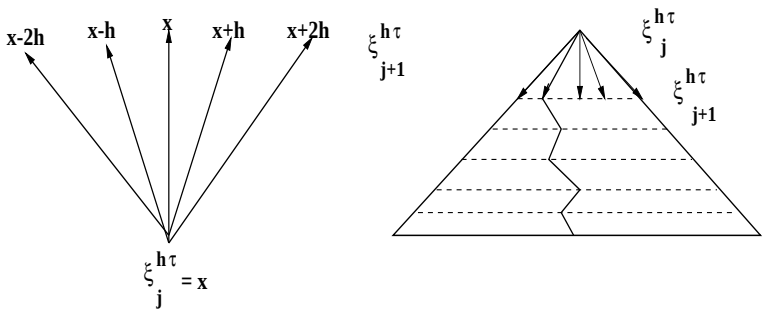

Figure 5: Markov chain as a tool to combine feedforward and feedbackwards neural networks

This results in

$$\lambda(b_{j+1} - b_j) \le 1, \quad \text{where} \quad b_{j+1} \le b_{j+2} \le 0 \le b_{j-1} \le b_j \tag{7.4}$$

(see Fig. 4 for interpretation). Consistency conditions for the general scheme (6.21) – (6.23) can also be obtained by using the methodology developed in [25]. Indeed, we assume that each jump of the associated Markov chain is allowed within a region (e.g., rectangular) with characteristic lengths τ and h such that $\tau < h$ (if necessary such a region can be easily refined in a way similar, for example, to the finite element methodology). If the Markov chain is in state x we associate this state with the position of the Markov chain, i.e. assume that $\xi^{h\tau}_j = x$. At the next moment of time $\tau \to \tau + \Delta\tau$ the Markov chain is assumed to be in one of the following states: $x - 2h$, $x - h$, x, $x + h$, or $x + 2h$ (see Fig. 5, left). This can be extended to the case of any arbitrary number of states in a straightforward manner. The table of transition probabilities from the old to a new state of the Markov chain associated with our scheme (6.21) – (6.23) can be constructed as follows.

New state (x_i)	Probability of transition (p_i)
$x - h$	$\tau/h(b_j - b_{j-1})$
$x + h$	$\tau/h(b_{j+2} - b_{j+1})$
x	$1 - \tau/h(b_{j+1} - b_j)$
$x - 2h$	τ/hb_{j-1}
$x + 2h$	$-\tau/hb_{j+2}$

We can now easily relate this jump of the Markov chain to the transition to the next time layer in our numerical approximation of the conservation law. Therefore, we are in a position to derive the explicit form of the local consistency condition for the family of schemes (6.21) – (6.23). Since

$$E_j^{\tau h|(x,d^j)} = \sum_{i=1}^{s} x_i p_i, \tag{7.5}$$

where s is the number of states allowed in the next time layer (in our case $s = 5$), after some simplifications this condition can be given in the form

$$\tau \sum_{k=j-1}^{j+2} b_k = o(\tau + h). \tag{7.6}$$

The sign is chosen to be positive due to the direction of computations (see Fig. 5, right).

For example, consider the family of schemes discussed in Section 5 (a subclass of the general family (6.21) – (6.23)). Note that in this case

$$\sum_{k=j-1}^{j+2} b_k = v^+[2(1+2\gamma_2)+\gamma_1+\gamma_2]+v^-(\gamma_3+\gamma_4)-|v|. \tag{7.7}$$

Take, for example, the limiters as $\gamma_1 = \gamma_2 = -1/2$ and $\gamma_3 = \gamma_4 = 1/2$. This reduces the consistency condition to

$$-v^+ + v^- - |v| = \sum_{k=j-1}^{j+2} b_k. \tag{7.8}$$

Note also that the resulting scheme in this case has the following form

$$u_i^{j+1} = u_i^j + \frac{\tau}{h}|v|\left[-\frac{3}{2}u_i + \frac{1}{2}(u_{i-1}+u_{i+1})\right] + \frac{\tau}{2h}[v^+u_{i-2} + v^-u_{i+2}]. \tag{7.9}$$

It is easy to conclude that if $v > 0$ (and hence $-2v = \sum_{k=j-1}^{j+2} b_k$) scheme (7.9) is reducible to

$$u_i^{j+1} = u_i^j + \frac{\tau}{2h}v[-3u_i^j + (u_{i+1}+u_{i-1}) + u_{i-2}], \tag{7.10}$$

whereas if $v < 0$ (and hence $\sum_{k=j-1}^{j+2} b_k = 0$) we have

$$u_i^{j+1} = u_i^j - \frac{\tau}{2h}v[-3u_i + (u_{i+1}+u_{i-1}) + u_{i+2}]. \tag{7.11}$$

The global consistency condition involves the velocity of the associated Markov chain (see, e.g. (5.14) and further details in [25]). Recall that for the family of schemes constructed in [25], the Markov chain velocity (that coincides with the velocity of the dynamic system/process when $n \to \infty$) is determined in terms of the flux limiters as follows

$$v_{\mathrm{MC}} = v^- - v^+ = -v. \tag{7.12}$$

This result is independent of the sign of v. It shows that the Markov chain evolves in the direction opposite to the evolution of the dynamic system. This allows us to represent an explicit form of the global consistency condition for the family of schemes (6.21) – (6.23) in the form:

$$\tau\left[3h|v| - \tau v^2\right] = o(\tau + h). \tag{7.13}$$

In conclusion, we note that, as follows from (7.6), the Lax-Wendroff scheme's consistency condition takes the form

$$\tau(b_j + b_{j+1}) = o(\tau+h) \quad \text{or} \quad \tau a = o(\tau+h), \tag{7.14}$$

which leads to the CFL-type stability condition

$$\frac{\tau a}{c(\tau+h)} \le 1 \tag{7.15}$$

where $c = \lim\limits_{\tau\to 0,\, h\to 0} \dfrac{2\tau v}{\tau+h}$ is a Landau constant participating in the definition of probabilistic characteristics of conservation laws [25]. Consistency conditions for other schemes considered in Section 6 can be obtained in a similar manner.

8 Conclusions

In this paper, a general framework for the analysis of a connection between the training of artificial neural networks via the dynamics of Markov chains and the approximation of conservation law models has been proposed. This framework allows us to demonstrate an intrinsic link between microscopic and macroscopic models for evolution via the concept of perturbed generalized dynamic systems. We have showed that mathematical models describing dynamics of network training can be treated effectively by using the concept of perturbed Markov chains associated with the original dynamic system. We have developed a general methodology allowing us to derive computational models for the dynamics of network training and to obtain constructive algorithms, as well as stability conditions for numerical approximations of conservation laws. We have demonstrated how this methodology can be applied to numerical approximations of conservation laws viewed here as Markov chain network training procedures. Our main results have been exemplified with several illustrative examples for which we have explicitly derived stability and consistency conditions.

Acknowledgments

I wish to thank my colleagues from Mathematics Departments at the University of South Australia for fruitful discussions on the topics of this paper. Support of the Mads Clausen Foundation is also gratefully acknowledged.

References

[1] O. Adetona, E. Garcia and L.H. Keel, A new method for the control of discrete nonlinear dynamic systems using neural networks, *IEEE Trans. Neural Networks* 11 (2000) 102–112.

[2] A. Agogino, K. Stanley, and R. Mikkulainen, Online interactive neuro-evolution, *Neural Processing Letters* 11 (2000) 29–37.

[3] J. Aguilar, Learning algorithm and retrieval process for the multiple classes random neural network model, *Neural Processing Letters*, 13 (2001) 81–91.

[4] M.S. Ahmed and S.H. Riyaz, Dynamic observers - a neural net approach, *J. of Intelligent and Fuzzy Systems* 9 (2000) 113–127.

[5] S.N. Balakrishnan, and V. Biega, Adaptive-critic-based neural networks for aircraft optimal control, *J. Guid. Control Dyn.* 19 (1996) 893–989.

[6] A.R. Barron, Universal Approximation bounds for superpositions of a sigmoidal function, *IEEE Trans. Inf. Theory* 39 (1993) 930-945.

[7] C. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

[8] J.B.D Cabrera and K.S. Narendra, Issues in the application of neural networks for tracking based on inverse control, *IEEE Trans. Autom. Control* 44 (1999) 2007–2027.

[9] A. Catala and C. Angulo, A comparison between the Tikhonov and the Bayesian approaches to calculate regularisation matrices, *Neural Processing Letters* 11 (2000) 185–195.

[10] G. J. Chaitin, Godel's Theorem and Information, *International Journal of Theoretical Physics* 22 (1982) 941-954.

[11] F. Corinto, M. Biey, M. Gilli, Non-linear coupled CNN models for multiscale image analysis, Int. J. of Circuit Theory and Applications 34 (1) (2006) 77-88.

[12] P. Costa and P. Larzabal, Initialization of Supervised Training for Parametric Estimation, *Neural Processing Letters* 9 (1999) 53–61.

[13] H. T. Croft, K. J. Falconer, and R. K. Guy, Unsolved Problems in Geometry, Springer-Verlag, New York, p. 3, 1991.

[14] G. Cybenko, Approximation by superposision of sigmoidal functions, *MCSS* 2 (1989) 303-314.

[15] B. DasGupta, B. Hammer, On approximate learning by multi-layered feedforward circuits, *Theoretical Computer Science* 348 (1) (2005) 95-127.

[16] M.O. Efe and O. Kaynak, Stabilizing and robustifying the learning mechanisms of ANNs in control engineering applications, *Int. J. of Intelligent Systems* 15 (2000) 365–388.
[17] D. Gawin, M. Lefik, and B.A. Schrefler, ANN approach to sorption hysteresis within a coupled hydro-thermo-mechanical FE analysis, *Int. J. Numer. Meth. Engng* 50 (2001) 299–323.
[18] R. B. Guenther and J. W. Lee, Partial Differential Equations of Mathematical Physics and Integral Equations. Prentice Hall, N.J., 1988.
[19] K. Hornik, M. Stinchcombe, and H. White, Multi-layer feedforward networks are universal approximators, *Neural Networks* 2 (1989) 359-366.
[20] L. Lapidus and G. F. Pinder, *Numerical Solution of PDEs in Science and Engineering*, John Wiley & Sons, 1999.
[21] A.U. Levin and K.S. Narendra, Control of nonlinear dynamical systems using neural networks. Part I, *IEEE Trans. Neural Networks* 4 (1993) 192–206 and Part II, *IEEE Trans. Neural Networks* 7 (1996) 30–42.
[22] L.-Z. Liao, A recurrent neural network for N-stage optimal control problems, *Neural Processing Letters* 10 (1999) 195–200.
[23] R.V.N. Melnik, A Hierarchy of Hyperbolic Macrodynamic Equations as a Model for Network Training, *Proceedings of the IEEE International Symposium on Information Theory*, Ulm, Germany, June 29 - July 4 (1997) 322.
[24] R.V.N. Melnik, On consistent regularities of control and value functions", *Numer. Func. Anal. and Optimiz.* 18 (1997) 401–426.
[25] R.V.N. Melnik, Dynamic system evolution and Markov chain approximation, *Discrete Dynamics in Nature and Society* 2 (1998) 7–39.
[26] R.V.N. Melnik, Deterministic and Stochastic Dynamics with Hyperbolic HJB-Type Equations", *Dynamics of Continuous, Discrete and Impulsive Systems. Series A: Mathematical Analysis* 10(3) (2003) 317–330.
[27] I. Müller and T. Ruggeri, *Extended Thermodynamics,* Springer-Verlag, 1993.
[28] K.S. Narendra, Neural networks for control: theory and practice", *Proc. of the IEEE* 84 (1996) 1385–1406.
[29] K.S. Narendra and J. Balakrishnan, Adaptive control using multiple models, *IEEE Trans. Automat. Contr.* 42 (1997) 171–187; and K.S. Narendra and S. Mukhopadhyaya, Adaptive control using neural networks and approximate models, *IEEE Trans. Neural Networks* 8 (1997) 475–485.
[30] K.S. Narendra and K. Parthasarathy, Identification and control of dynamical systems using neural networks, *IEEE Trans. Neural Networks* 1 (1990) 4–27.
[31] K.S. Narendra and C. Xiang, Adaptive control of discrete-time systems using multiple models, *IEEE Trans. Autom. Control* 45 (2000) 1669–1686.
[32] G. Puscasu and C. Milos, A new approach to multilayer neural networks, *Proc. Int. Conference on Technical Informatics: CONTI'96*, Timisoara (1996) 41–48.
[33] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, Springer, 2000.
[34] I. W. Sandberg and G.J.J. van Zyl, The spectral coefficients of the response of nonlinear systems to asymtotically almost periodic inputs, *IEEE Trans. on Circuits and Systems I - Fundamenta Theory and Applications* 48 (2001) 170–176.
[35] S. Santini and A. Del Bimbo, Properties of block feedback neural networks, *Neural Networks* 8(4) (1995) 579–596.
[36] E. Santos Jr. and J.D. Young, Probabilistic temporal networks: a unified framework for reasoning with time and uncertainty, *Int. J. of Approximate Reasoning* 20 (1999) 263–291.
[37] B. Sendhoff and M. Kreutz, A model for the dynamic interaction between evolution and learning, *Neural Processing Letters* 10 (1999) 181–193.
[38] C. W. Ulmer II et al, Computational neural network and the rational design of polymeric materials, *Computational and Theoretical Polymer Science* 8 (1998) 311–321.
[39] G. Wahba, Generalization and regularization of nonlinear learning systems, *in Handbook of Brain Theory and Neural Networks, 2nd ed., Ed. M. Arib*, 2000.
[40] J. Zhang et al, Prediction of polymer quality in batch polymerisation reactions using robust neural networks, *Chemical Engineering Journal* 69 (1998) 135–143.
[41] T. Zhang, S.S. Ge and C.C. Hang, Neural-based direct adaptive control for a class of general nonlinear systems, *Int. J. of Systems Science* 28 (1997) 1011–1020.